Part-2-Data-Preprocessing

December 21, 2021

1 Data Preprocessing

In this process, we load both the dataset and extract the user id and game id.After this we combine the data.

[ ]:
```
# Import libraries
import pandas as pd
import numpy as np

#Import Warnings
import warnings
warnings.filterwarnings("ignore")
```

1.1 Games data

Loading the game datafile for preprocessing.Evey row in data is different game.

[ ]:
```
game_data_frame = pd.read_json('gamesdata.json')
game_data_frame.head()
```

[ ]:
```
          publisher                                              genres  \
0         Kotoshiro      [Action, Casual, Indie, Simulation, Strategy]
1  Making Fun, Inc.                [Free to Play, Indie, RPG, Strategy]
2      Poolians.com  [Casual, Free to Play, Indie, Simulation, Sports]
3                                       [Action, Adventure, Casual]
4               NaN                                                NaN

                  app_name                     title  \
0     Lost Summoner Kitty      Lost Summoner Kitty
1                Ironbound                Ironbound
2  Real Pool 3D - Poolians  Real Pool 3D - Poolians
3                     2222                     2222
4           Log Challenge                      NaN

                                                 url release_date  \
0  http://store.steampowered.com/app/761140/Lost_…   2018-01-04
1  http://store.steampowered.com/app/643980/Ironb…   2018-01-04
2  http://store.steampowered.com/app/670290/Real_…   2017-07-24
```

```
3     http://store.steampowered.com/app/767400/2222/   2017-12-07
4  http://store.steampowered.com/app/773570/Log_C…          NaN

                                                 tags  discount_price  \
0      [Strategy, Action, Indie, Casual, Simulation]            4.49
1  [Free to Play, Strategy, Indie, RPG, Card Game…             NaN
2  [Free to Play, Simulation, Sports, Casual, Ind…             NaN
3                        [Action, Adventure, Casual]            0.83
4                    [Action, Indie, Casual, Sports]            1.79

                                        reviews_url  \
0  http://steamcommunity.com/app/761140/reviews/?…
1  http://steamcommunity.com/app/643980/reviews/?…
2  http://steamcommunity.com/app/670290/reviews/?…
3  http://steamcommunity.com/app/767400/reviews/?…
4  http://steamcommunity.com/app/773570/reviews/?…

                                             specs         price  \
0                                   [Single-player]          4.99
1  [Single-player, Multi-player, Online Multi-Pla…  Free To Play
2  [Single-player, Multi-player, Online Multi-Pla…  Free to Play
3                                   [Single-player]          0.99
4  [Single-player, Full controller support, HTC V…          2.99

   early_access        id         developer        sentiment metascore
0         False  761140.0         Kotoshiro              NaN       NaN
1         False  643980.0  Secret Level SRL  Mostly Positive       NaN
2         False  670290.0      Poolians.com  Mostly Positive       NaN
3         False  767400.0                               NaN       NaN
4         False  773570.0               NaN              NaN       NaN
```

We note that there are certain features which are lists, namely genres, tags and specs. These will be investigated further in the Data Exploration phase.

[ ]:
```
# Save as csv
game_data_frame.to_csv('gamesdata.csv')
```

1.2 User items Data

We now load the user items data, which has users as rows and details regarding items owned as columns.

[ ]:
```
# Load users/items data
user_data_frame = pd.read_json('data.json')
user_data_frame.head()
```

[ ]:
```
             user_id  items_count           steam_id  \
0  76561197970982479          277  76561197970982480
```

```
1            js41637          888  76561198035864384
2          evcentric          137  76561198007712560
3         Riot-Punch          328  76561197963445856
4              doctr          541  76561198002099488

                                                user_url  \
0  http://steamcommunity.com/profiles/76561197970…
1               http://steamcommunity.com/id/js41637
2             http://steamcommunity.com/id/evcentric
3            http://steamcommunity.com/id/Riot-Punch
4                 http://steamcommunity.com/id/doctr

                                                   items
0  [{'item_id': '10', 'item_name': 'Counter-Strik…
1  [{'item_id': '10', 'item_name': 'Counter-Strik…
2  [{'item_id': '1200', 'item_name': 'Red Orchest…
3  [{'item_id': '10', 'item_name': 'Counter-Strik…
4  [{'item_id': '300', 'item_name': 'Day of Defea…
```

We break the user and item data into seprate groups

[ ]:
```
# Extract items_count feat
numgames = user_data_frame[['user_id', 'items_count']]
numgames.head()
```

[ ]:
```
             user_id  items_count
0  76561197970982479          277
1            js41637          888
2          evcentric          137
3         Riot-Punch          328
4              doctr          541
```

[ ]:
```
# Save as csv
numgames.to_csv('numgames.csv')
```

We note that the `items` column appears to be a list of dictionaries. Let us look at it in further detail.

[ ]:
```
# Preview items column values for first user
# Restrict to first 2 items in dictionary

user_data_frame['items'][0][0:2]
```

[ ]:
```
[{'item_id': '10',
  'item_name': 'Counter-Strike',
  'playtime_forever': 6,
  'playtime_2weeks': 0},
 {'item_id': '20',
```

```
'item_name': 'Team Fortress Classic',
'playtime_forever': 0,
'playtime_2weeks': 0}]
```

Each game is represented by a dictionary with keys the game's `item_id`, `item_name`, `playtime_forever` and `playtime_2weeks`. The dictionaries are then storred in a list.

We will look to extract the `item_id`s into a seperate column. For now we will leave the playtime data but look to incorporate it later.

[ ]:
```
# Get all item_id for first user
gameids = [user_data_frame['items'][0][index]['item_id'] for index, _ in␣
↪enumerate(user_data_frame['items'][0])]
# Show first 10 item ids
gameids[:10]
```

[ ]:
```
['10', '20', '30', '40', '50', '60', '70', '130', '300', '240']
```

We will now generalize the above and create a column extracting the `item_id` from the dictionaries for each user.

[ ]:
```
# Create column with item ids
user_data_frame['item_id'] = user_data_frame['items'].apply(lambda x: [x␣
↪[index]['item_id'] for index, _ in enumerate(x)])
user_data_frame.head()
```

[ ]:
```
              user_id  items_count           steam_id  \
0  76561197970982479          277  76561197970982480
1            js41637          888  76561198035864384
2          evcentric          137  76561198007712560
3         Riot-Punch          328  76561197963445856
4              doctr          541  76561198002099488

                                           user_url  \
0  http://steamcommunity.com/profiles/76561197970…
1              http://steamcommunity.com/id/js41637
2            http://steamcommunity.com/id/evcentric
3           http://steamcommunity.com/id/Riot-Punch
4                http://steamcommunity.com/id/doctr

                                               items  \
0  [{'item_id': '10', 'item_name': 'Counter-Strik…
1  [{'item_id': '10', 'item_name': 'Counter-Strik…
2  [{'item_id': '1200', 'item_name': 'Red Orchest…
3  [{'item_id': '10', 'item_name': 'Counter-Strik…
4  [{'item_id': '300', 'item_name': 'Day of Defea…

                                             item_id
0  [10, 20, 30, 40, 50, 60, 70, 130, 300, 240, 38…
```

```
1  [10, 80, 100, 300, 30, 40, 60, 240, 280, 360, …
2  [1200, 1230, 1280, 1520, 220, 320, 340, 360, 3…
3  [10, 20, 30, 40, 50, 60, 70, 130, 80, 100, 300…
4  [300, 20, 50, 70, 130, 10, 30, 40, 60, 80, 100…
```

As the unique user `steam_id` is a large integer, we will replace it with a new `uid` counter, which starts at 0 and increments by 1 (like the index).

We will also only select the relevant columns for the purpose of building a user-item interactions matrix, namely the newly created user id `iud` and the `item_id`.

[ ]:
```
# Add a column with substitute user_id, counter
user_data_frame['uid'] = np.arange(len(user_data_frame))

# Take relevant columns
useritems = user_data_frame[['uid', 'item_id']]

# Check
useritems.head()
```

[ ]:
```
   uid                                             item_id
0    0  [10, 20, 30, 40, 50, 60, 70, 130, 300, 240, 38…
1    1  [10, 80, 100, 300, 30, 40, 60, 240, 280, 360, …
2    2  [1200, 1230, 1280, 1520, 220, 320, 340, 360, 3…
3    3  [10, 20, 30, 40, 50, 60, 70, 130, 80, 100, 300…
4    4  [300, 20, 50, 70, 130, 10, 30, 40, 60, 80, 100…
```

The next step is to explode the `item_id` into seperate rows, so each user-item interaction has it's own row.

[ ]:
```
# Explode item_ids into seperate rows
lst_col = 'item_id'
useritems = pd.DataFrame({col:np.repeat(useritems[col].values,␣
 ↪useritems[lst_col].str.len())
                                for col in useritems.columns.difference([lst_col])
                            }).assign(**{lst_col:np.concatenate(useritems[lst_col].
 ↪values)})[useritems.columns.tolist()]
useritems
```

[ ]:
```
          uid item_id
0           0      10
1           0      20
2           0      30
3           0      40
4           0      50
…         …      …
994981   9999    7670
994982   9999    8850
994983   9999    8870
```

```
994984  9999  409710
994985  9999  409720

[994986 rows x 2 columns]
```

As we are concerned with whether the game is owned, as opposed to ratings, we will add a binary column `owned` which will have 1s everywhere, as only items owned appear in the DataFrame.

[ ]:
```
# Add binary owned column
useritems['owned'] = np.ones(shape = useritems.shape[0])

# Check
useritems.head()
```

[ ]:
```
   uid item_id  owned
0    0      10    1.0
1    0      20    1.0
2    0      30    1.0
3    0      40    1.0
4    0      50    1.0
```

[ ]:
```
len(useritems)
```

[ ]:
```
994986
```

We note that we have over 5 million individual user-item relationships represented in our DataFrame.

We want to restrict ourselves to user-item relationships where the item is in the first `gamesdata` DataFrame to be able to extract relevant information such as genre.

We will ensure that the DataFrames can be merged on the game `id` feature by changing the type and column name.

[ ]:
```
# Change item_id to int
useritems['item_id'] = useritems['item_id'].astype(int)

# Rename column to match
useritems = useritems.rename(columns={'item_id': 'id'})
```

We can now merge the DataFrames.

[ ]:
```
# Merge useritems and games data dataframes
alldata = pd.merge(useritems, game_data_frame, on = 'id')
alldata.head()
```

[ ]:
```
   uid  id  owned publisher    genres        app_name           title  \
0    0  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
1    1  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
2    3  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
```

```
3    4  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
4   10  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike

                                                 url release_date  \
0  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
1  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
2  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
3  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
4  http://store.steampowered.com/app/10/CounterSt…   2000-11-01

                                               tags  discount_price  \
0  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
1  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
2  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
3  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
4  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN

                                        reviews_url  \
0  http://steamcommunity.com/app/10/reviews/?brow…
1  http://steamcommunity.com/app/10/reviews/?brow…
2  http://steamcommunity.com/app/10/reviews/?brow…
3  http://steamcommunity.com/app/10/reviews/?brow…
4  http://steamcommunity.com/app/10/reviews/?brow…

                                    specs price  early_access developer  \
0  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
1  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
2  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
3  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
4  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve

                sentiment metascore
0  Overwhelmingly Positive        88
1  Overwhelmingly Positive        88
2  Overwhelmingly Positive        88
3  Overwhelmingly Positive        88
4  Overwhelmingly Positive        88
```

[ ]:
```
def find_total_perc_missing (data_set):
    temp_missing_val = (data_set.isnull().sum()).sum()
    total_cel = np.product(data_set.shape)
    perc_missing_data=100 * (temp_missing_val/total_cel)
    return perc_missing_data
```

[ ]:
```
find_total_perc_missing(alldata)
```

[ ]:
```
8.517928035516288
```

[ ]:
```
def find_missing_value(data_set):
    percent_missing = data_set.isnull().sum() * 100 / len(data_set)
    missing_value_df = pd.DataFrame({'column_name': data_set.
 ↪columns,'percent_missing': percent_missing})
    missing_value_df=missing_value_df.sort_values('percent_missing',␣
 ↪ascending=False)

    return missing_value_df
```

[ ]:
```
find_missing_value(alldata).head(15)
```

[ ]:
```
                   column_name  percent_missing
discount_price  discount_price        99.872911
metascore            metascore        39.161547
publisher            publisher         3.329353
developer            developer         2.629344
genres                  genres         2.231307
release_date      release_date         1.914243
price                    price         1.908853
title                    title         1.779847
specs                    specs         0.450382
sentiment            sentiment         0.042882
tags                      tags         0.002036
reviews_url        reviews_url         0.000000
id                          id         0.000000
url                        url         0.000000
early_access      early_access         0.000000
```

[ ]:
```
alldata.shape
```

[ ]:
```
(834847, 18)
```

[ ]:
```
len(alldata)
```

[ ]:
```
834847
```

We have more than 8 lakh data

[ ]:
```
# Drop entries with no title
datawithnames = alldata.dropna(axis=0, subset=['title'])
datawithnames.head()
```

[ ]:
```
    uid  id  owned publisher    genres        app_name           title  \
0     0  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
1     1  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
2     3  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
3     4  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
4    10  10    1.0     Valve  [Action]  Counter-Strike  Counter-Strike
```

```
                                                 url release_date  \
0  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
1  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
2  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
3  http://store.steampowered.com/app/10/CounterSt…   2000-11-01
4  http://store.steampowered.com/app/10/CounterSt…   2000-11-01

                                               tags  discount_price  \
0  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
1  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
2  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
3  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN
4  [Action, FPS, Multiplayer, Shooter, Classic, T…             NaN

                                         reviews_url  \
0  http://steamcommunity.com/app/10/reviews/?brow…
1  http://steamcommunity.com/app/10/reviews/?brow…
2  http://steamcommunity.com/app/10/reviews/?brow…
3  http://steamcommunity.com/app/10/reviews/?brow…
4  http://steamcommunity.com/app/10/reviews/?brow…

                                    specs price  early_access developer  \
0  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
1  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
2  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
3  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve
4  [Multi-player, Valve Anti-Cheat enabled]  9.99         False     Valve

                 sentiment metascore
0  Overwhelmingly Positive        88
1  Overwhelmingly Positive        88
2  Overwhelmingly Positive        88
3  Overwhelmingly Positive        88
4  Overwhelmingly Positive        88
```

[ ]:
```
len(datawithnames)
```

[ ]:
```
819988
```

We will save this DataFrame as a `csv` file to conduct data exploration and gain insights.

[ ]:
```
# Save to csv
datawithnames.to_csv('mergeddata.csv')
```

Finally, let us extract the relevant columns for our user-item interactions matrix.

[ ]:
```
# Get relevant columns for recommendation engine
recdata = datawithnames[['uid','id','owned']]
recdata.head()
```

[ ]:

	uid	id	owned
0	0	10	1.0
1	1	10	1.0
2	3	10	1.0
3	4	10	1.0
4	10	10	1.0

[ ]:
```
# Save to csv
recdata.to_csv('recdata.csv')
```